Journal of Informatics, Electrical and Electronics Engineering
ISSN 2807-9507 (Media Online)
Vol 1, No 2, Desember 2021
Hal 54-61
https://djournals.com/jieee

Mendeteksi Otentikasi File Audio Menerapkan Metode Ripemd-128

Maria

Program Studi Teknik Informatika Universitas Budi Darma, Medan, Indonesia

Email : iyamaria450@gmail.com

Abstrak-File audio merupakan suatu sarana informasi dari satu orang ke orang lain atau dari suatu kelompok-kelompok lain. Perkembangan teknologi komputerisasi ini sudah sangat meninggkatkan. *File audio* sangat rentan terhadap penipuan, penyadapan maupun pencurian data oleh pihak-pihak yang tidak bertanggung jawab. Demi menjaga keamanan file audio dapat dilakukan dengan pemanfaatan teknik kriptografi. Kriptografi adalah sekumpulan teknik yang berguna untuk mengamankan informasi, Selain mengamankan informasi juga menjaga kerahasian dan keutuhan informasi tersebut.kriptografi digunakan untuk komunikasi penting seperti komunikasi dikalangan militer, diplomat, dan mata-mata. Namun saat ini kriptografi lebih *nonrepudiation*.RIPEMD-128 adalah suatu fungsi hash kriptografi yang dirancang untuk implementasi pada perangkat lunak dengan arsitektur 32 bit.

Kata Kunci: File Audio; Ripemd-128; Kriptografi

Abstract-Audio file is a means of information from one person to another or from a group to another. The development of computerized technology has greatly increased. Audio files are very vulnerable to fraud, eavesdropping and data theft by irresponsible parties. In order to maintain the security of audio files, this can be done by using cryptographic techniques. Cryptography is a collection of techniques that are useful for securing information. In addition to securing information, it also maintains the confidentiality and integrity of the information. Cryptography is used for important communications such as communications among the military, diplomats, and spies. But nowadays cryptography is more non-repudiation.RIPEMD-128 is a cryptographic hash function designed for implementation on software with 32 bit architecture.

Keywords: Audio Files; Ripemd-128; Cryptography

1. PENDAHULUAN

Pesan rahasia merupakan hasil penting yang butuh untuk dilindungi dan dijaga kerahasiaannya.Pesan rahasia merupakan salah satu rahasia dimana banyak orang yang ingin berusaha untuk mencari terlebih mengetahui isinya. Oleh karena itu maka tidak jarang muncul kejahatan-kejahatan yang dengan sengaja dilakukan oleh orang yang tidak bertanggung jawab. Dengan semakin banyaknya orang yang melakukan tindakan kriminal yang dengan sengaja melakukan pencurian data rahasia dan merusak data rahasia sehingga bisa merugikan pihak tertentu.

Penggunaan file audio pada pesan suara atau *voice note* memudahkan pengguna merekam suara dengan kemampuan *hand-free*. Ketika ingin merekam suara, pengguna cukup menekan tombol mikrofon lalu usap *(swipe)* ke atas hingga muncul gambar gembok terbuka. Dengan begitu, fungsi rekaman akan terkunci selama proses perekamanan berlangsung, pengguna bisa tetap menggulir *chat* di jendela obrolan yang terbuka. Sehingga, pengguna bisa melihat pesan mana yang harus dijawab. Setelah rekaman selesai, klik tombol kirim yang digambarkan ikon pesawat kertas ke arah kanan, untuk langsung mengirim pesan suara.

Rekaman file audio pada *voice* ini bisa dijadikan barang bukti ketika terjadi suatu tindakan pidana, maka kita tunjukan barang bukti tersebut kepihak yang berwajib. Apa bila barang bukti yang kita bawa tidak asli lagi atau sudah di manipulasi oleh sesorang yang tidak bertanggung jawab maka perlu untuk mendeteksi file audio tersebut.

RIPEMD 128 adalah salah satu contoh algoritma hash yang sering juga disebut dengan nama fungsi pembanding, fungsi penyusutan, intisari pesan, sidik jari, *message integrity check* (MIC) atau pemeriksa keutuhan pesan dan *manipulation detection code* (MDC) atau pendeteksi penyelenggaraan kode.

Menurut penelitian yang dilakukan oleh Abdurrasyid Aulia Rahman, dkk dalam penelitian yang berjudul **"**Perbandingan Algoritma Gosudarstvennyi Standard (Gost) Dan Ripe Message Digest (Ripemd) Pada Hashing Kode Booking Tiket Pesawat" menyimpulkan bahwa Nilai running time pada algoritma RIPE Message Digest (Ripemd) lebih kecil dibandingkan dengan nilai running time algoritma Gosudarstvennyi Standard (GOST) [1]. Artinya proses pada algoritma RIPE lebih cepat dibandingkan proses pada algoritma GOST. Penelitian lain yang dilakukan oleh Hans Dobbertin, dkk dalam penelitian **"**RIPE MD-128" menyimpulkan bahwa fungsi hash kriptografi untuk aplikasi seperti sidik jari digital dari pesan otentifikasi pesan dan derivasi kunci .

Salah satu solusi untuk mengatasi asli atau sudah dimanipulasi file audio tersebut dengan menggunakan metode RIPEMD 128. Fungsi hash sering juga disebut enkripsi satu arah, atau disebut juga *message diggest.* Fungsi hash digunakan untuk menjamin servis otentikasi dan integritas suatu pesan atau file. Suatu fungsi hash memetakan bit-bit *string* dengan panjang sembarang ke sebuah *string* dengan panjang tertentu misal n. Dengan domain D dan range R maka: Proses *hashing* merupakan proses pemetaan suatu input string menjadi *output* disebut ouput dari fungsi hash disebut nilai hash atau hasil hash.

RIPEMD 128 berbeda dari Fungsi hash hanya dengan rotasi *bitwise* tunggal dalam jadwal pesan dari fungsi kompresinya. ini dilakukan oleh NSA untuk memperbaiki kesalahan pada algoritma asli yang mengurangi keamanan kriptografiknya.

ISSN 2807-9507 (Media Online)
Vol 1, No 2, Desember 2021
Hal 54-61
https://djournals.com/jieee

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi, seperti kerahasiaan, keutuhan data, dan otentikasi entitas[2]. Banyak teknik dalam menjamin integritas data, salah satunya hash function. Fungsi Hash merupakan suatu tool yang penting dalam aplikasi kriptografi seperti sidik jari digital dari pesan,otentifikasi pesan dan penurunan kunci. Ada banyak algoritma yang dapat digunakan untuk hashing, dua algoritma diantaranya adalah algoritma Gosudarstvennyi Standard dan algoritma RIPE Message Digest.

2.2 Algoritma RIPEMD-128

Algoritma RIPEMD-128 *(RACE Integrity Primitives Evaluation Massage Di gest-128),* dirancang oleh Bart Preneel, Antoon Bosselaers dan Hans Dobbertin pada tahun 1996, merupakan fungsi hash kriptografi yang cepat yang dibuat untuk diimplementasikan pada software yang dijalankan pada mesin berarsitektur 32-bit. RIPEMD-128 adalah salah satu contoh algoritma hash yang sering juga disebut dengan nama fungsi pembanding, fungsi penyusutan, intisari pesan, sidik jari, *message integrity* check (MIC) atau pemeriska keutuhan pesan dan manipulation detection code (MDC) atau pendeteksi penyelenggaraan kode [5].

RIPEMD-128 adalah suatu fungsi hash kriptografi yang dirancang untuk implementasi pada perangkat lunak dengan arsitektur 32 bit. Desain utama dari fungsi hash ini ada dua proses komputasional yang berbeda dan saling idenpenden, dimana hasil dari kedua proses ini akan digabungkan pada akhir perhitungan dengan sebuah fungsi kompresi. Sesuai dengan namamya RIPEMD-128 menghasilkan 128 bit. Hal ini dimaksudkan untuk menyediakan level sekuritas yang lebih tinggi untuk 10 tahun yang akan datang. Sama hal nya dengan varian MD4 lainnya, RIPEMD-128 beroperasi pada fungsi ini adalah sebagai berikut:

1. Rotasi kiri *(left-rotation atau left-spin)* dari pesan masukan;
2. Operasi *bitwise Boolean (AND,NOT,OR,exclusive-OR);*
3. Penambahan dua buah *string* sepanjang modulo 2 pada nilai hash.

Dibawah ini adalah deskripsi langkah-langkah proses perhitungan nilai *hash* pada algoritma RIPEMD-128 :

4. Definisi fungsi, permutasi dan inisialisasi buffer

Sebelum proses perhitungan, terlebih dahulu didefinisikan fungsi-fungsi dan permutasi yang akan digunakan pada proses kompresi. Definisi kelima fungsi tersebut adalah:

a. F(x,y,z) = [(x) Xor (y) Xor (z)
b. G(x,y,z)= [(x) And (y) or [(not(x)) And (z)].
c. H(x,y,z)= [(x) or (not(y))] Xor (z))].
d. I(x,y,z)= [[(x) And(z)] or [(y) And (not(z))]].
e. J(x,y,z)= [(x) Xor [(y) or (not(z))]]

2.3 File Audio

Audio (suara) fenomena fisik yang dihasilkan oleh geteran suatu benda yang berupa sinyal analog dengan amplitudo yang berupa secara kontinyu terhadap waktu yang disebut frekunse (Nurasyiah, 201) Selama bergetar, perbedaan tekanan terjadi diudara sekitarnya, pola osilasi yang terjadi, dinamakan sebagai gelombang. Gelombang mempunyai pola sama yang berulang pada interval tertentu, yang disebut sebagai periodi

3. HASIL DAN PEMBAHASAN

3.1 Analisa

Analisa sistem adalah penguraian dari suatu informasi menuju kebagian-bagian komponennya dengan tujuan untuk mengevaluasi atau memecahkan permasalahan dan menghindari dan menghindari hambatan yang terjadi sehingga dapat menghasikan perbaikan pada sistem yang lama. Penulis akan menganalisa dan mengumpulkan semua kebutuhan yang diperlukan dalam mendeteksi keaslian file audio menggunakan metode Ripemd 128. Algoritma Ripemd 128 merupakan pengembangan dari algoritma Ripemd 160.Algoritma Ripemd 160 adalah algoritma hash yang merupakan peningkatan dari Ripemd 128. Peningkatan dan pengembangan ini dilakukan karena telah bnyak usaha-usaha kriptanalisis yang telah dilakukan terhadap Ripemd 128, dan berdasarkan informasi yang didapat maka diciptakanlah Ripemd 160.

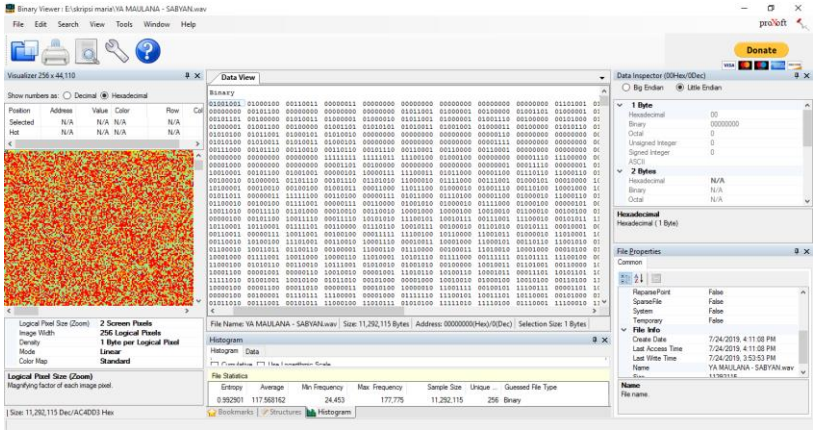
3.1.1 Contoh Kasus

Wav adalah format *audio* standart Microsoft dan IBM untuk personal computer (Pc), biasannya menggunakan coding Pulse Code Modulation (PCM). WAV adalah data tidak terkompres sehinnga seluruh sampel audio disimpan semuanya dihardisk. Software yang dapat menciptakan WAV dan *analog sound* misalnya adalah *Windows Sound Recorder*. File *audio* ini jarang digunakan di internet karena ukurannya relative besar dengan batasan maksimal untuk file WAV adalah 2GB, bisa dilihat pada gambar WAV tersebut.

Journal of Informatics, Electrical and Electronics Engineering
ISSN 2807-9507 (Media Online)
Vol 1, No 2, Desember 2021
Hal 54-61
https://djournals.com/jieee

Gambar 1. file audio WAV

Setelah melakukan proses WAV, selanjutnya masuk ke tahap pengambilan nilai Binary Viewer dari file audio tersebut. Bisa dilihat pada gambar Binary Viewer.



Gambar 2. Nilai Binary Viewer dari aplikasi Hasher Pro

Contoh kasus proses hitungan manual pada keaslian file audio menggunakan algoritma hash RIPEMD-128, dengan ekstensi jpeg, resolusi 1240 x 877 px. Untuk memudahkan proses analisa maka diambil sampel dari hasil citra keaslian file audio berukuran 5 x 5 pixel,dimana nilai pixel tersebut diambil menggunakan aplikasi Matlab. Nilai pixel citra keaslian file audio (sampel) dapat dilihat pada tabel berikut:

Tabel 1. Nilai RGB dari keaslian file audio

45	87	84	95	123
66	76	75	85	105
170	80	82	84	106
82	99	98	100	113
85	101	101	98	109

3.2 Penerapan Algoritma RIPEMD-128

Berikut ini langkah-langkah penerapan algoritma RIPEMD-128 untuk mendeteksi keaslian file audio, dimana sebelumnya terlebih dahulu dilakukan pengubahan nilai piksel dari keaslian file audio menjadi bilangan binner.

Tabel 2. ilai piksel RGB dalam biner

101101	1010111	1010100	1011111	1111011
1000010	1001100	1001011	1010101	1101001
10101010	1010000	1010010	1010100	1101010
1010010	1100011	1100010	1100100	1110001
1010101	1100101	1100101	1100010	1101101

a. Penambahan padding bit
Dari tabel nilai piksel RGB dalam biner diketahui bahwa panjang X=175 bit. Proses berikutnya adalah dengan menambahkan *padding* bit 1 dan sisanya 0 sejumlah *k*, dengan persamaan sebagai berikut :
$k = l + 1 \equiv 448 \bmod 512$
$k = 175 + 1 \equiv 448 \bmod 512$
$k = 176 \equiv 448 \bmod 512$
$k = 448 - 176$
$k = 272$ ukuran awal dari nilai piksel 272 adalah 100010000

Tabel 3. Penambahan Padding bit

101101	1010111	1010100	1011111	1111011	1000010	1001100	1001011
1010101	1101001	10101010	1010000	1010010	1010100	1101010	1010010
1100011	1100010	1100100	1110001	1010101	1100101	1100101	1100010
1101101	**10000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**
00000000	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**
00000000	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**
00000000	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**

b. Penambahan panjang *append*
Penambahan panjang *append*dilakukan dengan penambahan panjang pesan sebanyak 64 bit di akhir. Panjang pesan adalah 176 bit sehingga ditambahkan panjang *append* sebagai berikut :

Tabel 4. Penambahan panjang *append*

101101	1010111	1010100	1011111	1111011	1000010	1001100	1001011
1010101	1101001	10101010	1010000	1010010	1010100	1101010	1010010
1100011	1100010	1100100	1110001	1010101	1100101	1100101	1100101
1101101	**10000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**
00000000	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**
00000000	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**
00000000	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**00000000**	**100010000**

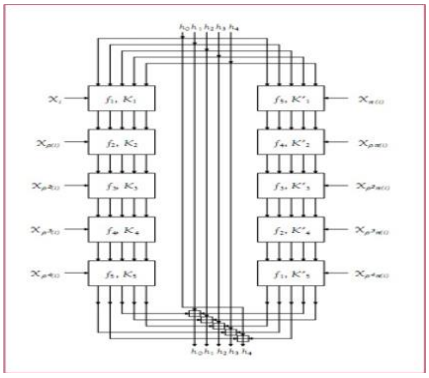
c. Parsing pesan (pengelompokan pesan)
Pada tahap parsing pesan, bagi X ke dalam 16 subblok 32 bit yaitu X(0)...X(15).

Tabel 5. *Parsing* Pesan

X_0	10110110101111010101001011111
X_1	11110111000010100111001001011
X_2	10101011101001101010101010000
X_3	10100101010100111010101010010
X_4	11000111100010111001001110001
X_5	10101011100101111001011100101
X_6	01101101100000000000000000000000
X_7	00000000000000000000000000000000
X_8	00000000000000000000000000000000
X_9	00000000000000000000000000000000
X_{10}	00000000000000000000000000000000
X_{11}	00000000000000000000000000000000
X_{12}	00000000000000000000000000000000
X_{13}	00000000000000000000000000000000
X_{14}	00000000000000000000000000000000
X_{15}	00000000000000000000000010101111

d. Inisialisasi nilai hash
Setelah proses *parsing* pesan maka langkah selanjutnya adalah inisialisasi nilai *hash* di mana nilai ini merupakan sebuah ketentuan yaitu :
H_0 : 67452301x;
H_1 : EFCDAB89x
H_2 : 98BADCFEx;
H_3 : 10325476x;
H_4 : C3D2EIFOx

e. Proses Kompresi
Proses kompresi merupakan inti dari proses algoritma RIPEMD-128, terdiri dari dua buah rantai fungsi. Diagram fungsi kompresi adalah sebagai berikut:



Gambar 3.3 Tahapan fungsi Kompresi

Dari gambar diatas,terlihat bahwa fungsi kompresi terdiri dari 5 ronde.Dalam satu ronde ada dua pemrosesan yang terjadi dan keduanya bekerja secara paralel dan bebas satu sama lain.Notasi notasi yang terlihat dari diagram X, f, dan K.

3.3 Implementasi

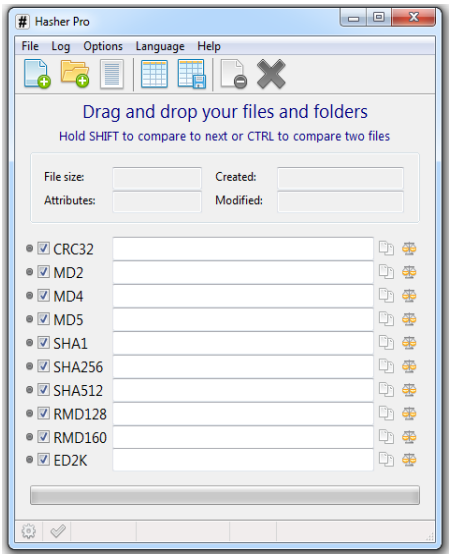
Implementasi Ripemd-128 adalah untuk mendeteksi keaslian *file audio*menggunakan *hasher pro* terdapat pada*file audio* yang asli dan satu yang palsu. *File audio* yang asli diberikan hasil *hash* dari Ripemd-128 ke piksel asli tersebut dengan pengerjaan sebanyak 5 tahap. Tahapan implementasi terdiri dari kebutuhan sistem komputer yang digunakan dan hasil pengujian *file audio*.

3.4 Pengujian

Padaproses pengujian orisinalitas file audio hasil asli dilakukan dengan alat bantu *tools Hasher Pro* untuk mendeteksi keaslian sebuah file berdasarkan nilai *hash* dari *file* audio tersebut.

Adapun langkah-langkah pengujian orisinalitas file audio hasil keaslian file audio dengan menggunakan *tools hasher pro* adalah sebagai berikut:

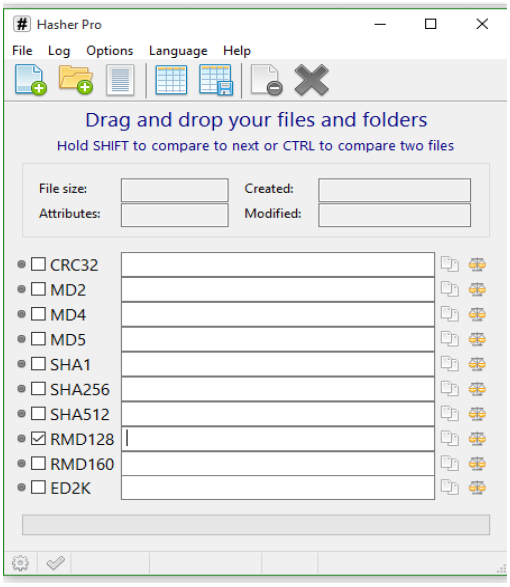
1. Jalankan *Hasher Pro*



Gambar 4 Tampilan *hasher pro*

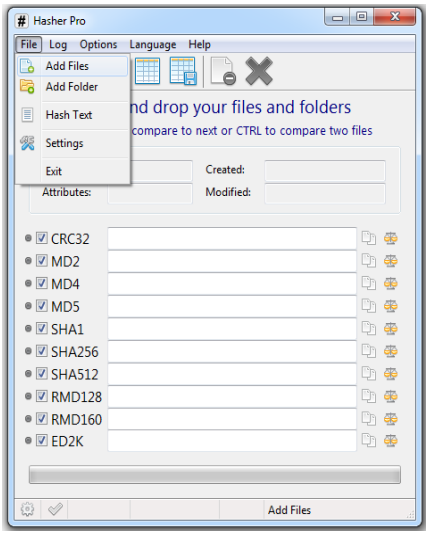
2. Pilih Metode RIPEMD-128

ISSN 2807-9507 (Media Online)
Vol 1, No 2, Desember 2021
Hal 54-61
https://djournals.com/jieee

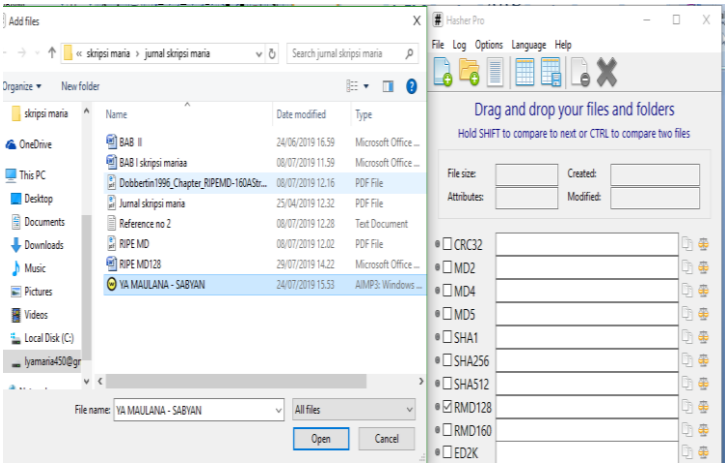


Gambar 5. Tampilan pilihan metode Ripemd-128

3. Pilih Add *File* untuk mengambil file audio hasil keaslian kemudian pilih *Open*.



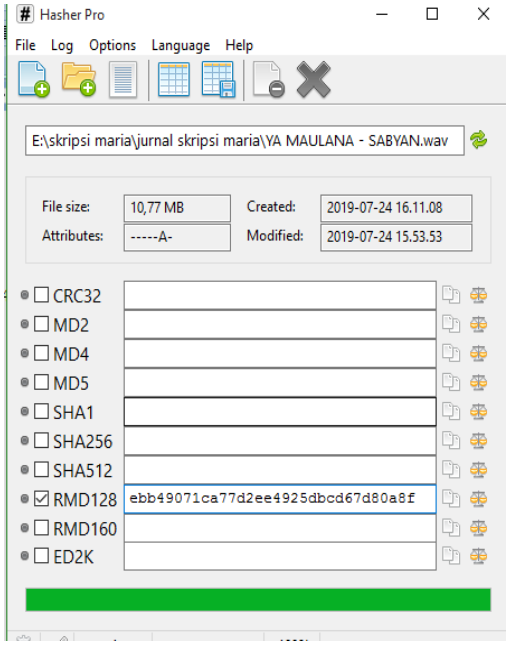
Gambar 6. Tampilan pengambilan gambar



Gambar 7 Tampilan pengambilan gambar

4. Tampilkan nilai *Hash*

ISSN 2807-9507 (Media Online)
Vol 1, No 2, Desember 2021
Hal 54-61
https://djournals.com/jieee



Gambar 4.4 Tampilan nilai *hash* metode Ripemd-128

3.5 Hasil Pengujian

Dengan menggunakan aplikasi *Hasher Pro* pada penggujian implementasi metode Ripemd-128 untuk mendeteksi *file audio* maka didapat sebuah hasil berikut ini.

Tabel 6. Hasil Hasher Pro

Parameter	File audio Asli	File audio Manipulasi	Nilai Hash file audio Asli	Nilai Hash file audio Manipulasi	Kesimpulan
Mengubah durasi	YA MAULANA - SABYAN-[AudioTrimmer.com] Tempo	YA MAULANA - SABYAN -[AudioTrimmer.com]Reverse	1a30b53a14a0e68eec597bdf3d1df18b	66ff426cfb29757d77092491e515ae1d	Terdapat perbedaan nilai hash antara file audio asli dengan file audio yang telah dimanipulasi dengan mengubah Durasifile audio.
Mengubah bitrate	YA MAULANA - SABYAN-[AudioTrimmer.com] Tempo	YA MAULANA - SABYAN -[AudioTrimmer.com]Reverse	1a30b53a14a0e68eec597bdf3d1df18b	a35f5608806ca59c0cf5ef636ce66c3b	Terdapat perbedaan nilai hash antara file audio asli dengan file audio yang telah dimanipulasi dengan mengubah Bitrate.
Cropping	YA MAULANA - SABYAN-[AudioTrimmer.com] Tempo	YA MAULANA - SABYAN -[AudioTrimmer.com]Reverse	1a30b53a14a0e68eec597bdf3d1df18b	a35f5608806ca59c0cf5ef636ce66c3b	Terdapat perbedaan nilai hash antarafile audio asli dengan file audio yang telah dimanipulasi dengan mengubah Cropping file audio.

4. KESIMPULAN

Adapun kesimpulan yang diperoleh dari penelitian ini adalah Proses deteksi file audio dilakukan dengan menerapkan metode Ripemd-128 dan telah berhasil melakukan proses mendeteksi keaslian *file audio* yang berformat WAV dan berjalan sesuai dengan teknik mendeteksi keasliannya. Deteksi otentikasi file audio dengan metode RIPEMD-128 menggunakan *tool Hasher Pro* dapat dilakukan dengan membandingkan nilai hash antara file video yang asli dengan file video yang telah dimanipulasi. Apabila nilai hash berbeda dari nilai hash file video asli maka dapat diperoleh keputusan bahwa file tersebut adalah palsu atau hasil manipulasi

REFERENCE

[1] Dobbertin, Hans c.s. (1996). RIPEMD160 A Strengthened Version of RIPEMD. German Information Security Agency.
[2] Masashi, Une c.s. (2006). IMES Discussion Paper Series – Year 2010 issues on cryptographic algorithms. Bank of Japan.
[3] Wang, Xiaoyun c.s. (2004). Collisions for Hash Functions MD4, MD5, HAVAL-128, and RIPEMD. Shandong University, Chinese Academy of Sciences, Shanghai Jiatong University.
[4] Preneel, Bart c.s. (1997). The Cryptographic Hash Function RIPEMD160. RSA Laboratories.
[5] Munir, R. 2004. Fungsi Hash Satu-Arah dan Algoritma MD5.